

# 1 Introduction

This paper presents a surround vehicle motion prediction algorithm for multi-lane turn intersections using a Long Short-Term Memory (LSTM)-based Recurrent Neural Network (RNN). More specific, it focused on improvement of in-lane target recognition and achieving human-like acceleration decisions at multi-lane turn intersections by introducing the learning-based target motion predictor and prediction-based motion predictor. A data-driven approach for predicting trajectory and velocity of surrounding vehicles on urban roads at multi-lane turn intersections is described. LSTM architecture, a specific kind of RNN capable of learning long-term dependencies, is designed to manage complex vehicle motions in multi-lane turn intersections. The results show that the forecaster improves the recognition time of the leading vehicle and contributes to the improvement of prediction ability

# 2 Previous Work

There are 3 main challenges to achieving fully autonomous driving on urban roads, which are scene awareness, inferring other drivers' intentions, and predicting their future motions. Researchers are developing prediction algorithms that can simulate a driver's intuition to improve safety when autonomous vehicles and human drivers drive together. To predict driver behavior on urban road, there are 3 categories for motion prediction model: (1) physics-based (2) maneuver-based; and (3) interaction-aware. Physics-based models are simple and direct, which only consider the states of prediction vehicles kinematically. The advantage is that it has minimal computational burden among the three types. However, it is impossible to consider interactions between vehicles. Maneuver-based models consider the driver's intention and classified them. By predicting the driver maneuver, the future trajectory can be predicted. Identifying similar behaviors in driving are able to infer different drivers' intentions which is stated to improve the prediction accuracy. However, it still an assistant to improve physics-based models. Recurrent Neural Network (RNN) is a type of approaches proposed to infer driver intention in this paper. Interaction-aware models can reflect interactions between surrounding vehicles, and predict future motions of detected vehicles simultaneously as a scene. While the prediction algorithm is more complex in computation which often used in offline simulations.

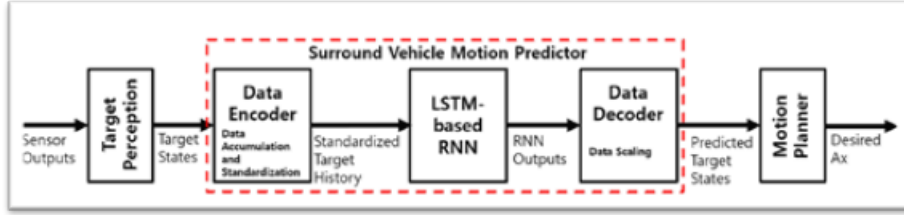


Figure 1: Overall architecture of the proposed surrounding target trajectory predictor

### 3 Motivation

It shows that less research focus on predicting trajectories at intersections. In addition, public data sets for analyzing driver behavior at intersections are not sufficient and not easy to collect. A model is needed to predict various movements of targets around multi-lane turning intersections. It is necessary to design a motion predictor that can be used for real-time traffic

### 4 Framework

The LSTM-RNN-based motion predictor comprises three parts: (1) a data encoder; (2) an LSTM-based RNN; and (3) a data decoder. depicts the architecture of the surrounding target trajectory predictor. The proposed architecture uses a perception algorithm to estimate the state of surrounding vehicles, which relies on six scanners. The output predicts the state of the surrounding vehicles and is used to determine the expected longitudinal acceleration in the actual traffic at the intersection.

### 5 LSTM-RNN based motion predictor

#### 5.1 Data

The real dataset is captured on urban roads in Seoul. The training model is generated from 484 tracks collected when driving through intersections in real traffic. The previous and subsequent states of a vehicle at a particular time can be extracted. After post-processing the collected data, a total of

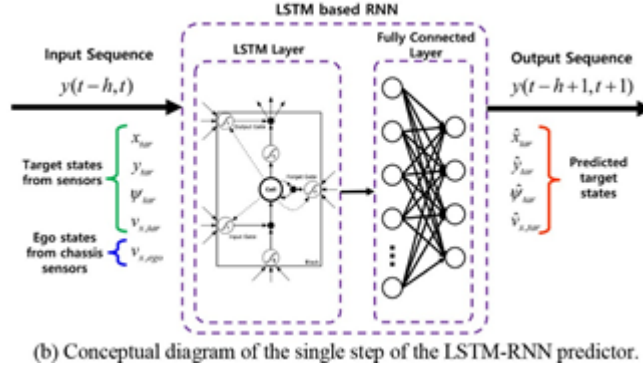


Figure 2: Conceptual diagram of the single step of the LSTM-RNN predictor

16,660 data samples were generated, including 11,662 training data samples and 4,998 evaluation data samples.

## 5.2 motion predictor

This article propose a data-driven method to predict the future movement of surrounding vehicles based on their previous movement. The motion predictor based on the LSTM-RNN architecture in this work only uses information collected from sensors on autonomous vehicles, as shown in the figure below. The contribution of the network architecture of this study is that the future state of the target vehicle is used as the input feature for predicting the field of view.

**Network architecture:** RNN is an artificial neural network, suitable for use with sequential data. RNN can also be used for time series data, where the pattern of the data depends on the time flow. It can contain feedback loops that allow activations to flow alternately in the loop. LSTM can avoid the problem of vanishing gradients by making errors flow backward without a limit on the number of virtual layers. This property prevents errors from increasing or declining over time, which can make the network training improperly. The figure below shows the various layers of LSTM-RNN and the number of units in each layer. This structure is determined by comparing the accuracy of 72 RNNs, which consist of a combination of four input sets and 18 network configurations.

**Input and output features:** In order to apply the motion predictor to the AV in motion, the speed of the data collection vehicle is added to

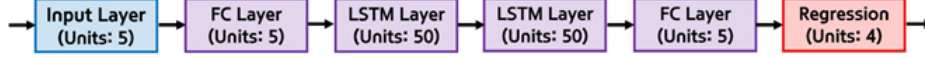


Figure 3: Depiction of the individual layers of the LSTM-RNN based motion predictor

the input sequence. The input sequence consists of relative X/Y position, relative heading angle, speed of surrounding target vehicles and speed of data collection vehicles. The output sequence is the same as the input sequence, such as relative position, heading and speed. **Encoder and decoder:** In this study, authors introduced an encoder and decoder that process the input from the sensor and the output from the RNN, respectively. The encoder normalizes each component of the input data to rescale the data to mean 0 and standard deviation 1, while the decoder denormalizes the output data to use the same parameters as in the encoder to scale it back to the actual unit. **Sequence length:** The sequence length of RNN input and output is another important factor to improve prediction performance. In this study, 5, 10, 15, 20, 25, and 30 steps of 100 millisecond sampling times were compared, and 15 steps showed relatively accurate results, even among candidates The observation time is very short.

## 6 motion planning based on surrounding vehicle motion prediction

In daily driving, experienced drivers will predict possible risks based on observations of surrounding vehicles, and ensure safety by changing behaviors before the risks occur. In order to achieve a human-like motion plan, based on the model predictive control (MPC) method, a prediction-based motion planner for autonomous vehicles is designed, which takes into account the driver's future behavior. The cost function of the motion planner is determined as follows:

$$J = \sum_{k=1}^{N_p} (x(k|t) - x_{ref}(k|t))^T Q (x(k|t) - x_{ref}(k|t)) + R \sum_{k=0}^{N_p-1} u(k|t)^2 + R_{\Delta\mu} \sum_{k=0}^{N_p-2} (u(k+1|t) - u(k|t))^2$$

where  $k$  and  $t$  are the prediction step index and time index, respectively;  $x(k|t)$  and  $x_{ref}(k|t)$  are the states and reference of the MPC problem, respectively;  $x(k|t)$  is composed of travel distance  $p_x$  and longitudinal velocity  $v_x$ ;  $x_{ref}(k|t)$  consists of reference travel distance  $p_{x,ref}$  and reference longitudinal velocity  $v_{x,ref}$ ;  $u(k|t)$  is the control input, which is the longitudinal acceleration command;  $N_p$  is the prediction horizon; and  $Q$ ,  $R$ , and  $R_{\Delta\mu}$  are the weight matrices for states, input, and input derivative, respectively, and these weight matrices were tuned to obtain control inputs from the proposed controller that were as similar as possible to those of human-driven vehicles. The constraints of the control input are defined as follows:

$$\begin{aligned}\mu_{min} &\leq \mu(k|t) \leq \mu_{max} \\ ||\mu(k+1|t) - \mu(k|t)|| &\leq S\end{aligned}$$

Determine the position and speed boundary based on the predicted state:

$$\begin{aligned}p_{x,max}(k|t) &= p_{x,tar}(k|t) - c_{des}(k|t) & p_{x,min}(k|t) &= 0 \\ v_{x,max}(k|t) &= \min(v_{x,ret}(k|t), v_{x,limit}) & v_{x,min}(k|t) &= 0\end{aligned}$$

Where  $v_{x,limit}$  are the speed limits of the target vehicle.

## 7 prediction performance analysis and application to motion planning

### 7.1 accuracy analysis

The proposed algorithm was compared with the results from three base algorithms, a path-following model with constant velocity, a path-following model with traffic flow and a CTRV model.

We compare those algorithms according to four sorts of errors, The  $x$  position error  $e_{x,T_p}$ ,  $y$  position error  $e_{y,T_p}$ , heading error  $e_{\theta,T_p}$ , and velocity error  $e_{v,T_p}$  where  $T_p$  denotes time  $p$ . These four errors are defined as follows:

$$\begin{aligned}e_{x,T_p} &= p_{x,T_p} - \hat{p}_{x,T_p} \\ e_{y,T_p} &= p_{y,T_p} - \hat{p}_{y,T_p} \\ e_{\theta,T_p} &= \theta_{T_p} - \hat{\theta}_{T_p} \\ e_{v,T_p} &= v_{T_p} - \hat{v}_{T_p}\end{aligned}$$

The proposed model shows a significantly less prediction errors compare to the based algorithms in terms of mean, standard deviation(STD), and root mean square error(RMSE). Meanwhile, the proposed model exhibits a bell shaped cure with a close to zero mean, which indicates that the proposed algorithm’s prediction of human divers’ intensions are relatively precise. On the other hand,  $e_{x,T_p}$ ,  $e_{y,T_p}$ ,  $e_{v,T_p}$  are bounded within reasonable levels. For instant, the three-sigma range of  $e_{y,T_p}$  is within the width of a lane. Therefore, the proposed algorithm can be precise and maintain safety simultaneously.

## 7.2 motion planning application

### 7.2.1 case study of a multi-lane left turn scenario

The proposed method mimic a human drivers better by simulating a human driver’s decision-making process. In a multi-lane left turn scenario, the proposed algorithm correctly predicted the trajectory of a target vehicle even the target vehicle was not following the intersection guide line.

### 7.2.2 statistical analysis of motion planning application results

The data is analysed from two perspectives, the time ot recognize the in-lane target and the similarity to human driver commands. In most of cases, the proposed algorithm detects the in-line target no late than based algorithm. In addition, the proposed algorithm only recognized cases later than the base algorithm did when the surrounding target vehicles first appeared beyond the sensors’ region of interest boundaries. This means that these cases took place sufficiently beyond the safety distance, and had little influence on determining the behavior of the subject vehicle.

In order to compare the similarities between the results form the proposed algorithm and human driving decisions, we introduced another type of error, acceleration error  $a_{x,error} = a_{x,human} - a_{x,cmd}$ . where  $a_{x,human}$  and  $a_{x,cmd}$  are the human driver’s acceleration history and the command from the proposed algorithm, respectively. The proposed algorithm showed more similar results to human drivers’ decisions than did the base algorithms. 91.97% of the acceleration error lies in the region  $\pm 1m/s^2$ . Moreover, base algorithm possesses limited ability to respond to different in-lane target behaviors in traffic flow. Hence, the proposed model is efficient and safety.